## **GPIO\_DRIVERS**

Main.c

|  |
| --- |
|  |

BML\_GPIO.c

(Double click to open)

|  |
| --- |
|  |

BML\_GPIO.h

(Double click to open)

|  |
| --- |
|  |

***GPIO Functions:-***

**CLOCK FUNCTION**

void gpio\_clk\_en(GPIO\_TypeDef \*port);

**GPIO CONFIGURATION FUNCTION**

void gpio\_config(GPIO\_TypeDef \*port, uint32\_t pinNumber, uint32\_t pinMode,uint32\_t pinSpeed, uint32\_t pinPUPD);

void gpio\_moder(GPIO\_TypeDef \*port, uint32\_t pinNumber, uint32\_t pinMode);

void gpio\_speed(GPIO\_TypeDef \*port, uint32\_t pinNumber, uint32\_t pinMode);

void gpio\_PUPD(GPIO\_TypeDef \*port, uint32\_t pinNumber, uint32\_t pinPUPD);

void gpio\_altfn(GPIO\_TypeDef \*port, uint32\_t pinNumber,uint32\_t alternate\_function);

**GPIO OPERATION FUNCTION**

void gpio\_write(GPIO\_TypeDef \*port, uint32\_t pinNumber, uint8\_t state);

void gpio\_toggle(GPIO\_TypeDef \*port, uint32\_t pinNumber);

uint8\_t gpio\_read(GPIO\_TypeDef \*port, uint32\_t pinNumber);

***How to use Functions:-***

* Define the name of port which you want as output(output as LED).
* Define the name of pin which you want as output(output as LED).
* Call the ‘gpio\_config’ function in ‘main’ function.
* The function will ask for 5 different variables.

“config\_gpio(port, pinNumber, pinMode, pinSpeed, pinPUPD)”.

* Port – replace it with the port name defined earlier.
* pinNumber – replace it with the pin name defined earlier.
* pinMode – As we saw in earlier example, there are 4 types of Mode. We will use the OUTPUT\_MODE. It is defined in the header file.
* pinSpeed – 3 different speed defined in header file. Use LOW\_SPEED this time.
* pinPUPD – Pin Pull-up or Pull down. We can either disable it using DI\_PUPD or enable either Pull-up using EN\_PU or Pull-down using EN\_PD.
* We have 3 operation function, we will use only two in this example.
* First is ‘gpio\_write(port, pinNumber, state)’.
* Port and pin in similar to what we used earlier, the ‘state’ is to drive the pin high using ‘HIGH’ or low using ‘LOW’.
* Other is ‘gpio\_toggle(port, pinNumber)’.
* It will switch between high and low whenever triggered.

***How the function works?***

1. **GPIO\_Config function** – This Funtion is what do the settings of the gpio by calling the other functions like clock enabling, gpio\_moder, gpio\_speed, gpio\_PUPD.
2. **GPIO\_clk\_en function** – It takes the port as parameter and using if -else cases it turns on the GPIO clock of particular port. The *GPIOx\_CLK\_EN is just the definition of (RCC->AHBENR |= RCC\_AHBENR\_GPIOAEN)* defined in the header file to make it look more easy to read.
3. **GPIO Moder function –** In moder register, we have to use two bit spaces to configure one Pin. Means to configure the pin 2, we have to shift 2 bits. So instead their is an array which constains integers with multiple of ‘2’. If we want to select the mode as output which is 01 on bit ‘2’ then we can call the 2nd element of array which stores the value 0x04. 01 << 4(01 left shifted 4 times). It is hard to remember shfting 4 or 6 for the pin but easy if it is written like 01 << PINPOS\_2B[2]. Here 2 means PIN 2. Also as this function is to select the mode as INPUT, OUTPUT, ANALOG, ALTERNATE instead of writing 00,01,10,11 their is a definition of them in header file which healping in understanding it better .OUTPUT\_MODE << PINPOS\_2B[2] is more readable than the about snipent.
4. **GPIO SPEED function –** This is similar to what we did in moder register, except this time we are doing it on OSPEED register.
5. **GPIO\_PUPD function -** This is similar to what we did in moder register, except this time we are doing it on PUPD register.
6. **GPIO Write and toggle function –** There are two registers to set the pin output high or low. One is useful for write high/low, other is to toggle or set/reset it. For gpio\_write, we are using BSRR register.In this register lower 15 bits is for setting the pin high and the upper 15 bits is to set the pin low or reset state. If the state is HIGH the lower half of the bit is set to one which Set the pin high, if we set the state to LOW then the upper half of the bit is set which makes the PIN low. Ther other is ODR register which if have a ‘1’ will set the Pin high and if ‘0’ then pin will be low. Using Xor, we can make it 1 and then 0 and then 1 i.e. toggling.